

REMARKS

The October 15, 2004, Final Office Action ("Office Action") continues to reject all the claims in this application (1-18) under 35 U.S.C. § 102(e) and maintains that this application is fully anticipated by the teachings of U.S. Patent No. 6,576,849 (Ludovici et al.). Claims 1, 2, and 11 have been amended to more particularly point out and distinctly claim the subject matter of applicants' invention. For the reasons hereinafter set forth, applicants respectfully submit that Claims 1-18, particularly as amended, are not anticipated by the teachings of Ludovici et al.

Prior to discussing in detail the claims, as amended, and why applicants believe that all of the claims in this application are allowable, a brief description of applicants' invention and a brief description of the teachings of Ludovici et al. are provided. The following descriptions of applicants' invention and Ludovici et al. are not provided to define the scope or interpretation of any of the claims of this application. Instead, these descriptions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Applicants' Invention

Applicants' invention is directed to a method, computer-readable medium, and apparatus for dynamically configuring a server computer. In one exemplary embodiment, a global configuration file is utilized to configure a server computer on a per request basis. When a request is received at a server computer, the server computer **dynamically** configures itself utilizing the contents of the global configuration file and **dynamically changing information not contained in the global configuration file such as, for example, the current location of the server and the domain from which the request was received**. The global configuration file contains configuration settings that may or may not be utilized by the server computer in configuring itself. To determine if particular configuration settings should be utilized by the server computer in configuring itself, the server computer parses rules **also contained in the global configuration file**. The rules are applied using information not contained in the configuration file **such as, for example, the current location of the server and the domain from which the request was received**. Configuration settings are determined based upon information provided with the request or from information maintained in a registry of the Web server computer and not **originally available** in the setting or configuration files themselves.

If a rule is evaluated as true, configuration settings associated with the rule are used by the server computer when configuring itself. The server computer can then respond to the request utilizing its **dynamic** configuration. Thus, in one form, the invention is directed to a

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESSsm
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

method for **dynamically** configuring a server computer. The method comprises, in response to receiving a **request for a resource** located at the server computer, **analyzing said request for a resource**. The method further comprises, **based on said analysis, dynamically determining** one or more configuration settings **from information contained in said request for a resource**. The method also comprises evaluating a configuration rule using the configuration settings to determine whether the configuration rule is satisfied. In response to determining that the configuration rule is satisfied, the method further comprises adding one or more configuration settings associated with the configuration rule to the configuration settings to create new configuration settings. Finally, the method includes configuring the server computer based upon the new configuration settings.

In another form, the invention is directed to computer-controlled apparatus comprising a central processing unit, a memory, a network interface, a storage device, and a global configuration file stored on the storage device. The global configuration file comprises one or more configuration rules and one or more configuration settings associated with each configuration rule. The central processing unit is operative to execute instructions stored in the memory that cause the computer-controlled apparatus to take certain actions in response to **receiving a request for a resource accessible to the computer-controlled** apparatus via the network interface. The executed instructions cause the computer-controlled apparatus to **analyze the request for a resource and, based on the analysis, dynamically determine** one or more configuration settings **from information contained in the request for a resource**. The executed instructions also cause the computer-controlled apparatus to evaluate one of the configuration rules to determine whether the configuration rule is satisfied and, in response to determining that the configuration rule is satisfied, add the configuration settings associated with the configuration rule to a configuration setting file stored in the memory or on the storage device. Finally, the executed instructions cause the computer-controlled apparatus to respond to the request for a resource using the configuration settings.

Ludovici et al.(U.S. Patent No. 6,567,849)

Ludovici et al. is directed to a system and method for configuring and administering multiple instances of Web servers. More specifically, Ludovici et al. is directed to a system and a method for managing a plurality of instances of Internet connection servers. A Web browser displays and interacts with a plurality of HTML **forms** and corresponding common gateway interface binary programs that are provided selectively for creating and deleting instances of servers, associating a configuration file with a server instance, changing server instance start-up parameters, and starting, ending, and restarting server instances (Col. 2, lines 51-59). The forms

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

are designed and implemented so that an Internet connection server can be **configured directly with no reference to information outside of the configuration file**. A configuration file for an Internet connection server is built by selecting and entering values upon these forms (Col. 4, lines 3-7). For example, a plurality of instances of Internet connection servers (ICSs) exist at a site and need to be managed. A **form** is presented by the administration server listing this plurality of other servers and the user selects the instance (that is, the other server) he needs to administer (start, stop, change, add, delete). **Upon selecting a server and an action, a page is presented or displayed at the browser of the administration server which allows further configuration. The content of that page depends upon the selected action** (Col. 4, lines 14-22).

In summary, Ludovici et al. is directed to a system that includes a plurality of forms for entering configuration values. **Ludovici et al. does not teach or even remotely suggest configuring a server based upon configuration settings developed as a result of receiving a request for a resource located at the server computer.**

The Claims

As noted above, the Office Action rejected Claims 1-18 under 35 U.S.C. § 102(e) as being fully anticipated by Ludovici et al. As also noted above, applicants respectfully disagree with this rejection.

Claims 1, 10, and 11, the first set of claims referred to in the Office Action, as amended, read as follows:

1. A method for dynamically configuring a server computer, comprising:
 - in response to receiving a **request for a resource** located at said server computer **analyzing said request for a resource;**
 - based on said analysis, dynamically determining one or more configuration settings from information contained in said request for a resource;**
 - evaluating a configuration rule using said configuration settings to determine whether said configuration rule is satisfied;**
 - in response to determining that said configuration rule is satisfied, **adding one or more configuration settings associated with said configuration rule to said configuration settings to create new configuration settings; and**
 - configuring said server computer based upon said new configuration settings.** (Emphasis added.)

10. A computer-readable medium having computer executable instructions for performing the method of Claim 1.

11. A computer-controlled apparatus, comprising:
a central processing unit;
a memory;
a network interface;
a storage device; and

a global configuration file stored on said storage device comprising one or more configuration rules and one or more configuration settings associated with each configuration rule; and wherein

said central processing unit is operative to execute instructions stored in said memory which, in response to receiving a request for a resource accessible to said computer-controlled apparatus via said network interface, cause said computer-controlled apparatus to:

- (i) **analyze said request for a resource;**
- (ii) **based on said analysis, dynamically determine one or more configuration settings from information contained in said request for a resource;**
- (iii) **evaluate one of said configuration rules to determine whether said configuration rule is satisfied;**
- (iv) in response to determining that said configuration rule is satisfied, **add said configuration settings** associated with said configuration rule to a configuration settings file stored in said memory or on said storage device; and
- (v) respond to said request for a resource using said configuration settings. (Emphasis added.)

Applicants submit that Ludovici et al. clearly does not anticipate the subject matter of Claims 1, 10, and 11, particularly as amended. Ludovici et al. clearly does not anticipate "in response to receiving a request for a resource located at said server computer, analyzing said request for a resource." Nor does Ludovici et al. anticipate "based on said analysis, dynamically determining one or more configuration settings from information contained in said request for a resource." While Ludovici et al. does purportedly disclose an administration Internet connection server, Ludovici et al. discloses nothing whatsoever with regard to applying a request **for a resource** located at the server computer and/or using the results of the analysis to dynamically determine one or more configuration settings from information contained in the request for a resource. Ludovici et al. does not determine one or more configuration settings from information contained in a request for a resource, much less **dynamically** determine such configuration settings from such a request. Dynamically is well understood in this art to mean occurring immediately and concurrently.

Nor does Ludovici et al. disclose "evaluating a configuration rule using said [dynamically determined] configuration setting to determine whether said configuration rule is satisfied." (Bracketed material added for clarity.) While Ludovici et al. purportedly discloses "rules," here the similarity ends. Ludovici et al. clearly does not disclose evaluating a configuration rule using configuration settings, much less dynamically determined configuration settings, to determine whether a configuration rule is satisfied. In this regard, Col. 8, lines 19-36, of Ludovici et al. read as follows:

In operation, referring to FIG. 5, after a start command 320 is received, in step 322 * ADMIN instance file 318 is read and appropriate parameters are passed to the main thread of the program. In step 324, the main thread then reads the configuration files 314, 316, builds a rules list from them, and then connects to the HTTP (and perhaps HTTPS) ports. In steps 326 and 328, the main thread of server 310 listens (waits) on the HTTP port, and the SSL listener thread listens (waits) on the HTTPS port for work to arrive (that is, requests from browser 304), and then passes these requests to a non-busy worker thread 330 from a managed pool of worker threads. Worker thread 330 is so named since it does all the work, reading, parsing and comparing the request against the rules list, and then acting on the rules that are encountered. When processing by worker thread 330 is complete, it returns information of some kind to browser 304, which may be the resource 312 requested, output of a CGI application 306, or an error message.

The foregoing portion of Ludovici et al. clearly does not disclose: in response to receiving a request for a resource located at a server computer, analyzing said request for a resource; based on said analysis, dynamically determining one or more configuration settings from information contained in said request for a resource; or evaluating a configuration rule using the dynamically determined configuration settings to determine whether the configuration rule is satisfied. Nor does Ludovici et al. disclose, in response to determining that a configuration rule is satisfied, adding one or more configuration settings associated with the configuration rule to the configuration settings to create a new configuration setting, and configuring a server computer based upon the new configuration settings. This subject matter is not disclosed in FIGURES 12 and 13; Col. 35, line 18, through Col. 36, line 39; or Col. 41, line 42, through Col. 42, line 10, or any other place in Ludovici et al., as far as applicants have been able to determine. As a result, applicants respectfully submit that Claims 1, 10, and 11 (which include parallel recitations, albeit in apparatus form) are clearly not anticipated by Ludovici et al. and, thus, are clearly allowable.

Applicants further submit that Claims 2-9 and 12-15, all of which are dependent from Claims 1 or 11, directly or indirectly, are also allowable. Many of these claims include

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

recitations that clearly are not taught or suggested by Ludovici et al., even when considered in the abstract. More importantly, when the claims are considered in combination, the overall combination is clearly not taught or suggested by Ludovici et al. For example, Claims 2 and 12, which depend from Claims 1 and 11, respectively, read as follows:

2. The method of Claim 1, further comprising:
dynamically determining whether an additional configuration rule remains to be tested; and

in response to determining that an additional configuration rule remains to be tested,

(i) evaluating said additional configuration rule with said configuration settings to determine if said additional configuration rule is satisfied,

(ii) in response to determining that said additional configuration rule is satisfied, adding one or more configuration settings associated with said additional configuration rule to said configuration settings to create new configuration settings, and

(iii) configuring said server computer based upon said new configuration settings.

12. The computer-controlled apparatus of Claim 11, wherein said central processing unit is further operative to execute instructions stored in said memory which cause said computer-controlled apparatus to:

determine whether one of said configuration rules is untriggered; and

in response to determining that one of said configuration rules has not been triggered,

(i) evaluating said untriggered configuration rule to determine if said untriggered configuration rule is satisfied,

(ii) in response to determining that said untriggered configuration rule is satisfied, adding said configuration settings associated with said untriggered configuration rule to said configuration settings file stored in said memory or on said storage device to create a new configuration settings file; and

(iii) responding to said request for a resource using said new configuration settings.

Applicants submit that the subject matter of Claims 2 and 12 is clearly not taught at the location cited in paragraph 4 of the Office Action. This is particularly true when the subject matter of Claims 2 and 12 is considered in combination with the subject matter of Claims 1 and 11, the claims from which these claims depend. Thus, applicants submit that Claims 2 and 12, and all of the other claims in this application dependent on Claims 1 and 11, are allowable for reasons in addition to the reasons why Claims 1 and 11 are allowable, particularly when considered in combination with the claims from which these claims depend.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

Claims 16-18 are data structure claims. Independent Claim 16 reads as follows:

16. A computer-readable medium having stored thereon a data structure, comprising:
- a. a first data field comprising a configuration rule; and
 - b. a second data field comprising one or more configuration settings associated with said configuration rule used to configure a server computer if said configuration rule is satisfied.

Dependent Claims 17 and 18 read as follows:

17. The computer-readable medium of Claim 16, wherein said configuration settings comprise at least one setting name and a setting value associated with said setting name.

18. The computer-readable medium of Claim 17, further comprising:
(c) a third data field comprising at least one required configuration setting.

The citations to Ludovici et al. set forth in paragraphs 12, 13, and 14 on page 5 of the Office Action (Figure 5; Col. 8, lines 19-36; Table 1; Col. 8, line 47, through Col. 9, line 58; and Figures 11 and 13) disclose only that data is read from one or more sources of data. These citations do not even remotely suggest how the data might be structured. Ludovici et al. discloses nothing regarding data structures, much less the data structures recited in Claims 16-18.

Specifically, FIGURE 5 of Ludovici et al. is a flow diagram. Flow diagrams illustrate program flow and/or control, not data structures. The Col. 8, lines 19-36, text only suggests that certain data are read and used without describing a data structure. Table 1, Col. 8, line 47, through Col. 9, line 58, presents pseudocode that describes how certain data may be read, but does not disclose the structure of the data that is read. FIGURES 11 and 13 show user interfaces. These FIGURES disclose how data is entered by a user, not how data is stored after entry or the structures used to store the data.

The paragraph 12-14 citations only describe how data is used. The citations do not describe any data structures, much less the particular data structures recited in Claims 16-18.

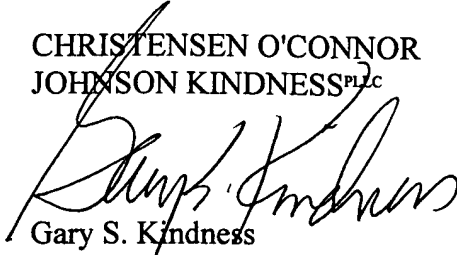
In summary, applicants respectfully submit that independent Claims 1, 11, and 16 are not anticipated by Ludovici et al. and, thus, are allowable. Applicants further submit that the remaining claims, all of which are dependent on Claims 1, 11, and 16 directly or indirectly, are also not anticipated by Ludovici et al., and are, therefore, also allowable. Consequently, favorable action passing this application to issue is respectfully solicited.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

If the Examiner has any questions, the Examiner is invited to contact applicants' attorney at the number set forth below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLC}


Gary S. Kindness

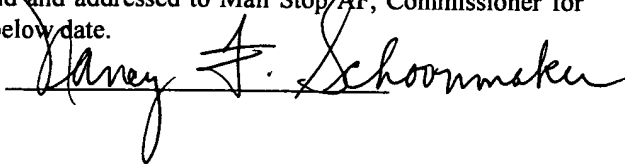
Reg. No. 22,178

Direct Dial No. 206.695.1702

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date:

1/18/05


Nancy F. Schoonmaker

GSK/MFM:jam/nfs